Reduced Convolutional Neural Networks for image recognition in professional appliances

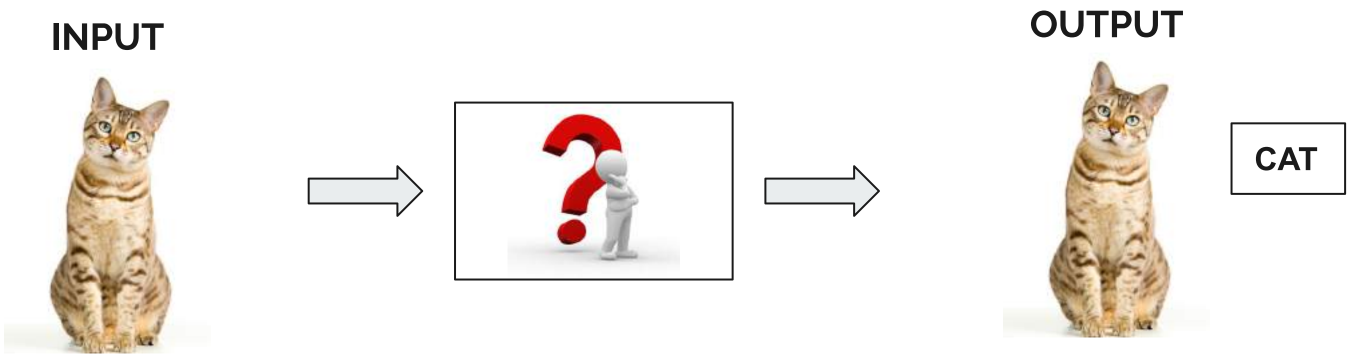
17 December 2021, SISSA, Trieste
RAMSES: Reduced order models; Approximation theory; Machine Learning; Surrogates, Emulators and Simulators.

Laura Meneghetti, N. Demo, G. Rozza

mathLab, Mathematics Area, SISSA International School for Advanced Studies, Trieste, Italy

Image Recognition

Image Recognition is a subcategory of Computer Vision, that represents a set of methods for analyzing images in order to identify the elements within an image.



Convolutional Neural Networks

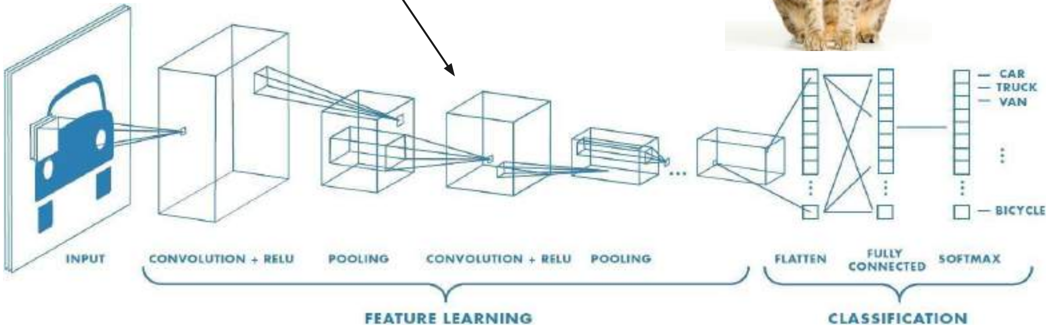
A **Convolutional Neural Network** (ConvNet/CNN) is a Deep Learning algorithm which can take as input an image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other.

INPUT

Convolutional Neural Network

OUTPUT

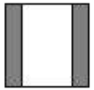
CAT



An example of feature extraction: vertical lines

CONVOLUTION

$$
\begin{bmatrix}
10 & 10 & 10 & 0 & 0 & 0 \\
10 & 10 & 10 & 0 & 0 & 0 \\
10 & 10 & 10 & 0 & 0 & 0 \\
10 & 10 & 10 & 0 & 0 & 0 \\
10 & 10 & 10 & 0 & 0 & 0 \\
10 & 10 & 10 & 0 & 0 & 0
\end{bmatrix}
*
\begin{bmatrix}
1 & 0 & -1 \\
1 & 0 & -1 \\
1 & 0 & -1
\end{bmatrix}
=
\begin{bmatrix}
0 & 30 & 30 & 0 \\
0 & 30 & 30 & 0 \\
0 & 30 & 30 & 0 \\
0 & 30 & 30 & 0
\end{bmatrix}
$$

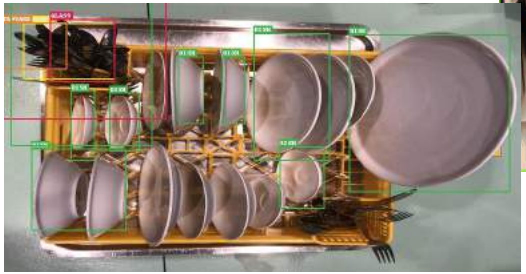


- With different filters you can extract different features: horizontal lines, 45° lines and so on…

Note: The filters are not selected and placed inside the convolutional neural network. All the values in those matrices are let to be parameters and learned automaitcally from data.

Image Recognition: case studies

- ➔ Smart cooking for ovens
- ➔ Recognition of different types of food placed inside a fridge
- ➔ Recognition of different types of crockeries for efficient washing in a dishwasher
- ➔ ...



IMPROVE AND SIMPLIFY THE LIFE OF THE FINAL CUSTOMER

Electrolux PROFESSIONAL

The Research Hub™ By ELECTROLUX PROFESSIONAL

Implementation steps

Choice of the net

There exists a lot of CNNs that have already been implemented in order to solve the problem of Image Recognition:
- VGG: **VGG-16**
-ResNet
-AlexNet
-....

Construction of the dataset

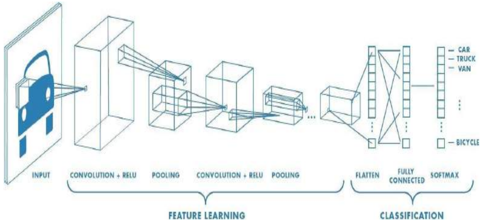
CIFAR10

Custom Dataset

Train and test the model

Training process: phase during which the net is learning to classify the objects
Testing process: phase in which we are testing the model (see correctness of predictions)

Practical application in a professional appliance



Convolutional Neural Network

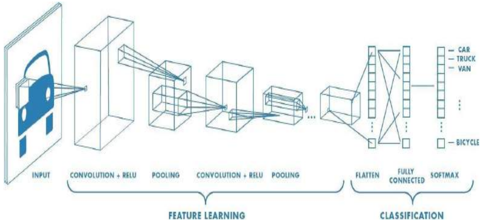
DIMENSIONALITY PROBLEM!!

Embedded system with memory constraints

Our net require 56 Mb of memory storage, but we do not have so much space!

HOW CAN WE SOLVE THIS?

Practical application in a professional appliance



Convolutional Neural Network

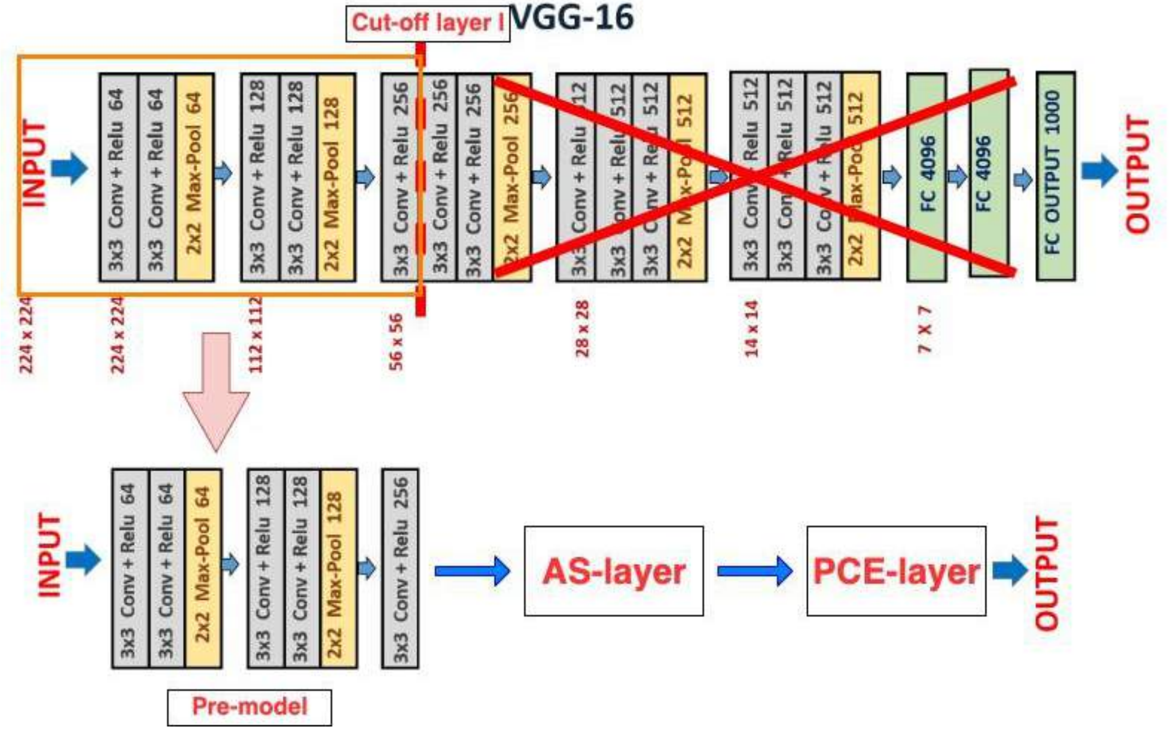
DIMENSIONALITY PROBLEM!!

Our net require 56 Mb of memory storage, but we do not have so much space!

Embedded system with memory constraints

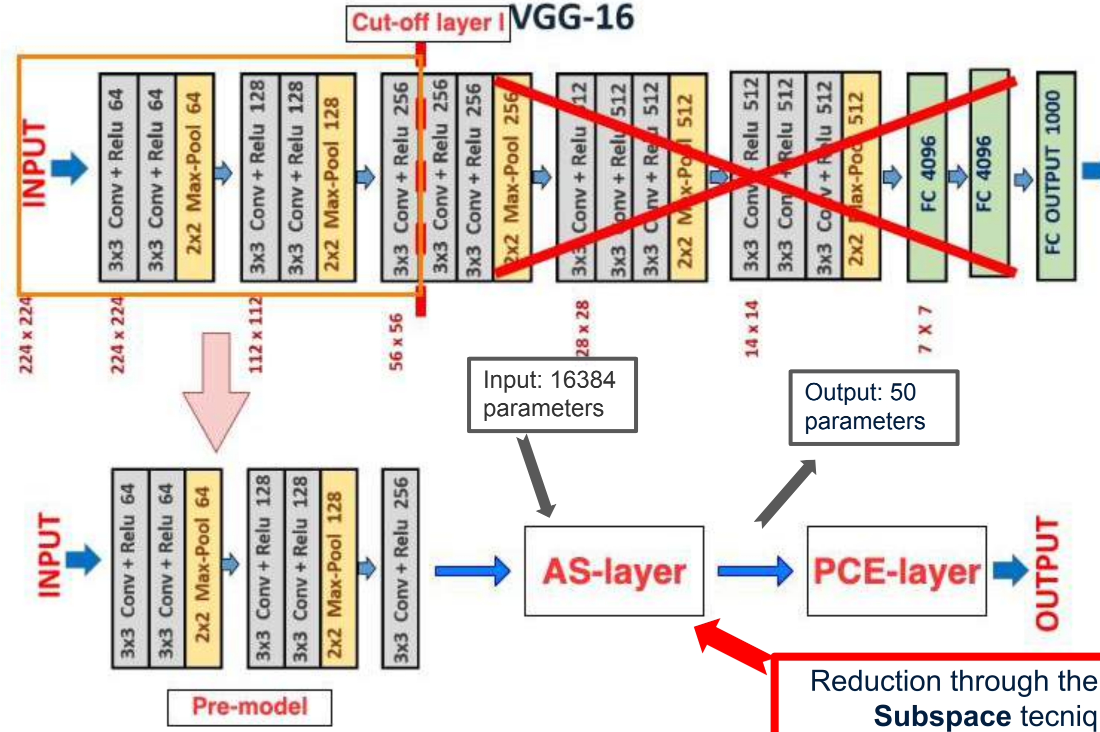
DIMENSIONALITY REDUCTION OF THE NEURAL NETWORK

Reduced Convolutional Neural Network: idea



References:
Chunfeng Cui, Kaiqi Zhang, Talgat Daulbaev, Julia Gusak, Ivan Oseledets, and Zheng Zhang. "*Active Subspace of Neural Networks: Structural Analysis and Universal Attacks*", (2020)
SIAM Journal on Mathematics of Data Science (SIMODS)

Reduced Convolutional Neural Network: idea



Reduction through the **Active Subspace** tecnique

Input : $(\mathbf{x}_i^0, \hat{\mathbf{y}}_i)$ for $i = 1, \dots, m$

$$\mathcal{CNN} \equiv f_L \circ f_{L-1} \circ \cdots \circ f_1$$
$$\mathcal{CNN}^l_{pre} = f_l \circ f_{l-1} \circ \cdots \circ f_1$$
$$\mathcal{CNN}^l_{post} = f_L \circ f_{L-1} \circ \cdots \circ f_l$$

$$\mathbf{z} = \mathbf{V}_1^T \mathbf{x} \in \mathbb{R}^r$$

$\mathbf{V}_1$ active subspace

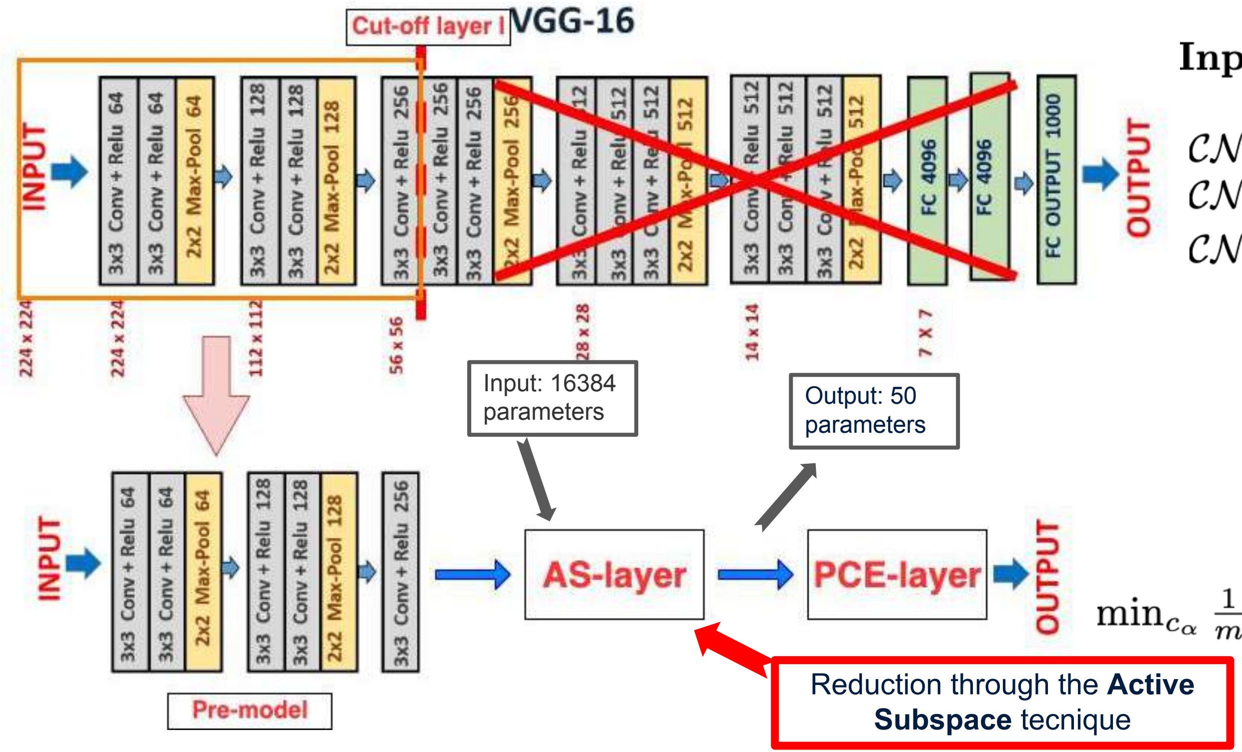
$$\mathbf{C} = \mathbb{E}[\nabla c(\mathbf{x}) \nabla c(\mathbf{x})^T] = \int (\nabla_{\mathbf{x}} c)(\nabla_{\mathbf{x}} c)^T \rho d\mathbf{x}$$

where $c(\mathbf{x}) = loss(\mathcal{CNN}^l_{post}(\mathbf{x}))$

$\mathbf{C} = \mathbf{V} \mathbf{\Lambda} \mathbf{V}^T$ eigenvalue decomposition

$\mathbf{V} = [\mathbf{V}_1, \mathbf{V}_2]$

Reduced Convolutional Neural Network: idea



Input : $(\mathbf{x}_i^0, \hat{\mathbf{y}}_i)$ for $i = 1, \dots, m$

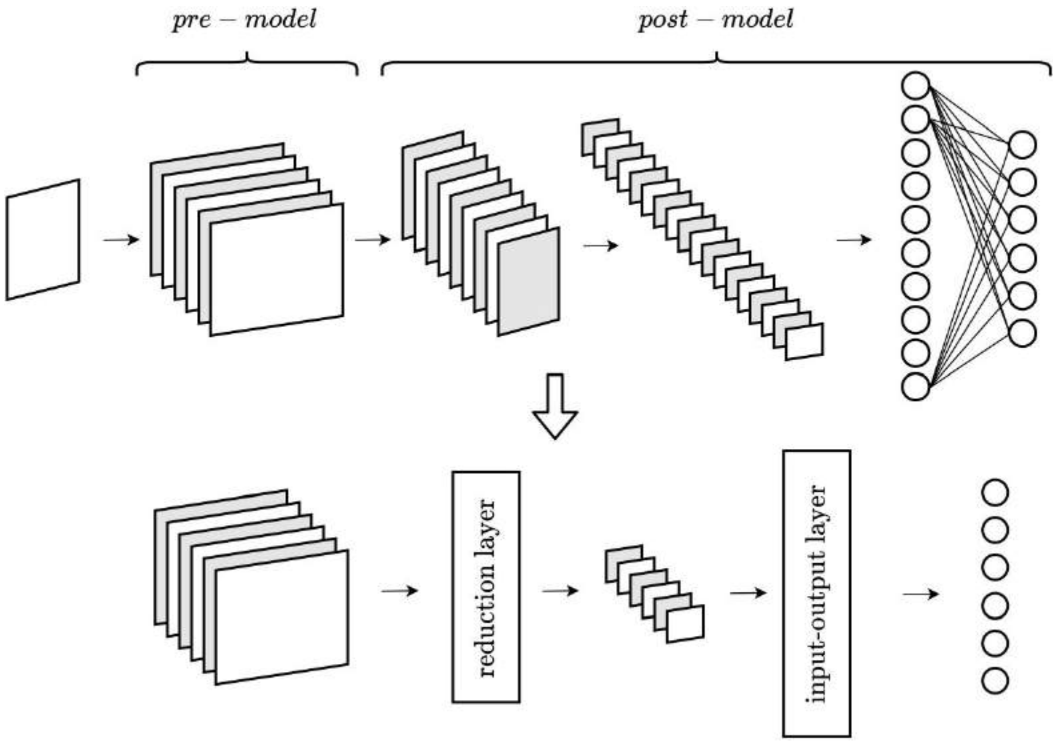
$$\mathcal{CNN} \equiv f_L \circ f_{L-1} \circ \dots \circ f_1$$
$$\mathcal{CNN}^l_{pre} = f_l \circ f_{l-1} \circ \dots \circ f_1$$
$$\mathcal{CNN}^l_{post} = f_L \circ f_{L-1} \circ \dots \circ f_l$$

$$\mathbf{z} = \mathbf{V}_1^T \mathbf{x} \in \mathbb{R}^r$$

$$\min_{c_\alpha} \frac{1}{m} \sum_{j=1}^m \|\hat{\mathbf{y}}_j - \sum_{|\boldsymbol{\alpha}|=0}^p \mathbf{c}_{\boldsymbol{\alpha}} \boldsymbol{\Phi}_{\boldsymbol{\alpha}}(\mathbf{z}_j)\|^2$$

Reduced Convolutional Neural Network: general framework

"A Dimensionality Reduction Approach for Convolutional Neural Networks", Meneghetti L, Demo N. Rozza G., arXiv:2110.09163

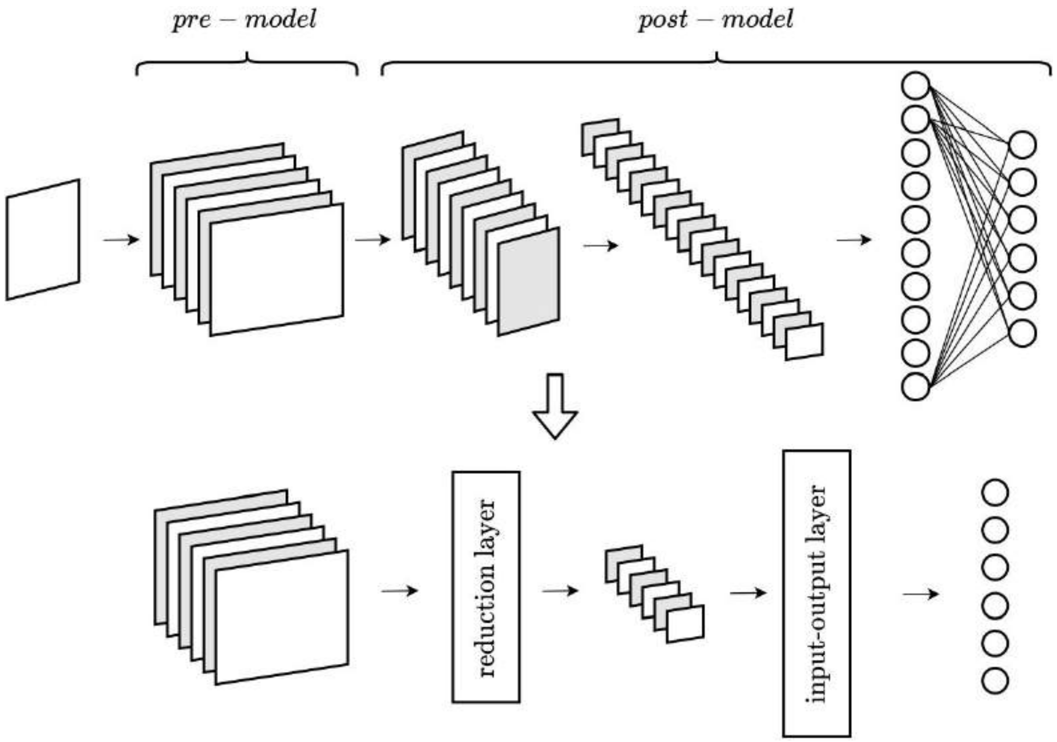


General Convolutional Neural Network

Reduced Convolutional Neural Network

- **Pre-model**
- **Reduction Layer**
- **Input-Output Layer**

Reduced Convolutional Neural Network: general framework



"A Dimensionality Reduction Approach for Convolutional Neural Networks", Meneghetti L, Demo N. Rozza G., arXiv:2110.09163

General Convolutional Neural Network

Reduced Convolutional Neural Network

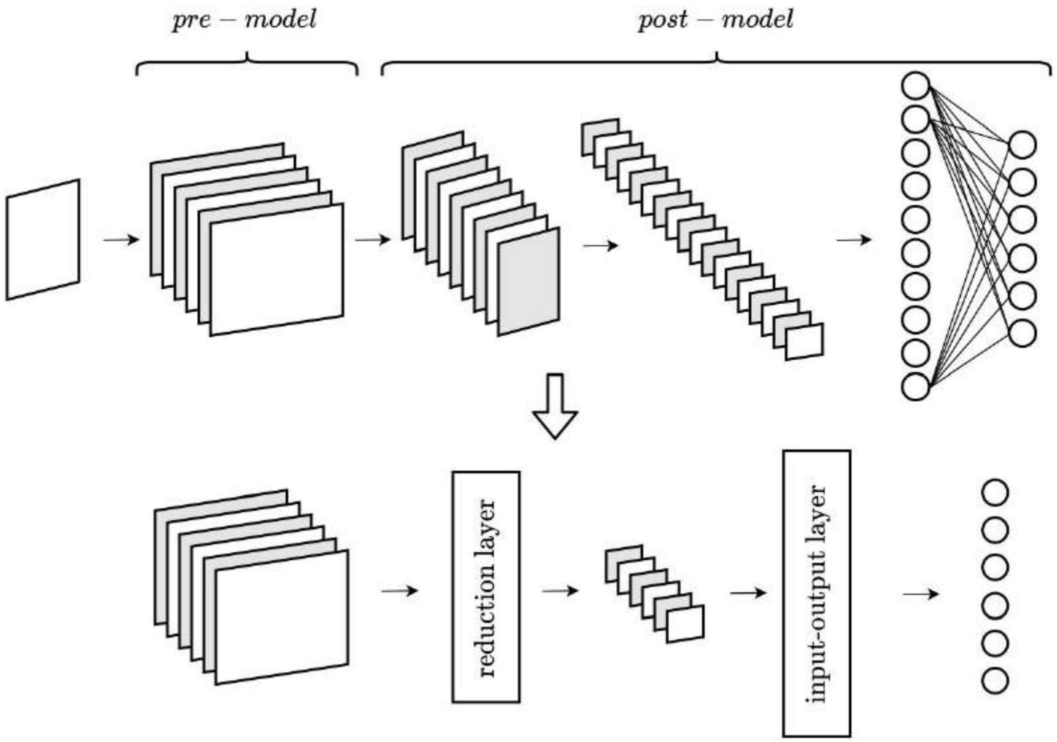
- **Pre-model**
- **Reduction Layer**

Proper Orthogonal Decomposition

Active Subspaces

- **Input-Output Layer**

Reduced Convolutional Neural Network: general framework



"A Dimensionality Reduction Approach for Convolutional Neural Networks", Meneghetti L, Demo N. Rozza G., arXiv:2110.09163

General Convolutional Neural Network

Reduced Convolutional Neural Network

- **Pre-model**
- **Reduction Layer**
- **Input-Output Layer**

Feed Forward Neural Network

Polynomial Chaos Expansion

Reduced Convolutional Neural Network

Algorithm 3.1 A pseudo-code for the construction of the reduced convolutional neural network

Input: A dataset with m input samples $\mathcal{D}_0 = \{\mathbf{x}_j^0\}_{j=1}^m$, a convolutional neural network $\mathcal{CNN}$, $\{\hat{y}_j\}_{j=1}^m$ real output of the $\mathcal{CNN}$, reduced dimension r, index of the cut-off layer l

1: $\mathcal{CNN}_{pre}^l, \mathcal{CNN}_{post}^l = \text{splitting\_net}(\mathcal{CNN}, l)$
2: $\mathbf{x}^l = \mathcal{CNN}_{pre}^l(\mathbf{x}^0)$
3: $\mathbf{z}^l = \text{reduce}(\mathbf{x}^l, r)$
4: $y = \text{in\_out\_map}(\mathbf{z}^l, \hat{y})$
5: Training of the constructed reduced net

Output: Reduced Net $\mathcal{CNN}^{red}$

Active Subspace $\mathbf{z} = \mathbf{V}_1^T\mathbf{x} \in \mathbb{R}^r$

Proper Orthogonal Decomposition

$\mathbf{z} = \mathbf{U}_r^T\mathbf{x} \in \mathbb{R}^r$ where $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$ snapshot matrix

Reduced Convolutional Neural Network

Algorithm 3.1 A pseudo-code for the construction of the reduced convolutional neural network

Input: A dataset with m input samples $\mathcal{D}_0 = \{\mathbf{x}_j^0\}_{j=1}^m$, a convolutional neural network $\mathcal{CNN}$, $\{\hat{y}_j\}_{j=1}^m$ real output of the $\mathcal{CNN}$, reduced dimension r, index of the cut-off layer l

```
CNN^l_pre, CNN^l_post = splitting_net(CNN, l)
x^l = CNN^l_pre(x^0)
z^l = reduce(x^l, r)
y = in_out_map(z^l, ŷ)
Training of the constructed reduced net
```

$\mathcal{CNN}^l_{pre}, \mathcal{CNN}^l_{post} = \text{splitting\_net}(\mathcal{CNN}, l)$

$\mathbf{x}^l = \mathcal{CNN}^l_{pre}(\mathbf{x}^0)$

$\mathbf{z}^l = \text{reduce}(\mathbf{x}^l, r)$

$y = \text{in\_out\_map}(\mathbf{z}^l, \hat{y})$

Training of the constructed reduced net

Output: Reduced Net $\mathcal{CNN}^{red}$

Feed Forward Neural Network

Polynomial Chaos Expansion

$$\min_{c_\alpha} \frac{1}{m} \sum_{j=1}^m \|\hat{\mathbf{y}}_j - \sum_{|\boldsymbol{\alpha}|=0}^p \mathbf{c}_{\boldsymbol{\alpha}} \boldsymbol{\Phi}_{\boldsymbol{\alpha}}(\mathbf{z}_j)\|^2.$$

$$y^j = \sum_{i=1}^{n_1} w_{ji}^{(2)} z^{(1),i} = \sum_{i=1}^{n_1} w_{ji}^{(2)} \sigma\left(\sum_{m=1}^{r} w_{im}^{(1)} z^m\right), \qquad j = 1, \ldots, n_{classes}$$

Results CIFAR10

CIFAR10 Dataset: a computer-vision dataset used for object recognition. It consists of 60000 32x 32 colour images divided in 10 non-overlapping classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck.

Network	Accuracy		Storage (Mb)			Time	
VGG-16	77.98%		56.15			46 h	
	Epoch 0	Epoch 10	Pre-M	AS/POD	PCE/FNN	Init	Train
AS+PCE (5)	13.52%	82.01%	2.12	3.12	0.05	43 min	4.5 h
AS+FNN (5)	33.06%	80.43%	2.12	3.12	0.0047	5 h	4.5 h
POD+FNN (5)	62.16%	80.24%	2.12	3.12	0.0047	79 min	5 h
AS+PCE (6)	14.42%	84.69%	4.37	3.12	0.05	49 min	5.5 h
AS+FNN (6)	33.76%	82.13%	4.37	3.12	0.0047	5 h	4.5 h
POD+FNN (6)	63.84%	83.93%	4.37	3.12	0.0047	83 min	5 h
AS+PCE (7)	4.25%	85.60%	6.62	0.78	0.05	35 min	5.5 h
AS+FNN (7)	75.66%	86.03%	6.62	0.78	0.0047	1.5 h	5 h
POD+FNN (7)	80.17%	87.45%	6.62	0.78	0.0047	12 min	5 h

Results CIFAR10

CIFAR10 Dataset: a computer-vision dataset used for object recognition. It consists of 60000 32x 32 colour images divided in 10 non-overlapping classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck.

Network	Accuracy		Storage (Mb)			Time	
VGG-16	77.98%		56.15			46 h	
	Epoch 0	Epoch 10	Pre-M	AS/POD	PCE/FNN	Init	Train
AS+PCE (5)	13.52%	82.01%	2.12	3.12	0.05	43 min	4.5 h
AS+FNN (5)	33.06%	80.43%	2.12	3.12	0.0047	5 h	4.5 h
POD+FNN (5)	62.16%	80.24%	2.12	3.12	0.0047	79 min	5 h
AS+PCE (6)	14.42%	84.69%	4.37	3.12	0.05	49 min	5.5 h
AS+FNN (6)	33.76%	82.13%	4.37	3.12	0.0047	5 h	4.5 h
POD+FNN (6)	63.84%	83.93%	4.37	3.12	0.0047	83 min	5 h
AS+PCE (7)	4.25%	85.60%	6.62	0.78	0.05	35 min	5.5 h
AS+FNN (7)	75.66%	86.03%	6.62	0.78	0.0047	1.5 h	5 h
POD+FNN (7)	80.17%	87.45%	6.62	0.78	0.0047	12 min	5 h

Results custom dataset

Custom Dataset: composed of 3448 32x32 colour images organized in 4 classes: 3 non-overlapping classes and a mixed one, composed of pictures with objects of different categories present at the same time.

Network	Accuracy		Storage (Mb)			Time	
VGG-16	95.65%		56.14			22 min	
	Epoch 0	Epoch 10	Pre-M	AS/POD	PCE/FNN	Init	Train
AS+PCE (5)	29.03%	95.21%	2.12	3.12	0.02	2 min	10 min
AS+FNN (5)	94.63%	94.92%	2.12	3.12	0.0021	12.5 min	12 min
POD+FNN (5)	96.52%	96.66%	2.12	3.12	0.0021	28 sec	11.5 min
AS+PCE (6)	29.75%	95.79%	4.37	3.12	0.02	2.5 min	10 min
AS+FNN (6)	94.92%	95.36%	4.37	3.12	0.0021	12.5 min	12.5 min
POD+FNN (6)	96.23%	96.37%	4.37	3.12	0.0021	33 sec	13 min
AS+PCE (7)	28.59%	94.05%	6.62	0.78	0.02	1.5 min	11 min
AS+FNN (7)	94.34%	94.63%	6.62	0.78	0.0021	4.5 min	13 min
POD+FNN (7)	96.37%	96.52%	6.62	0.78	0.0021	33 sec	14 min

Results custom dataset

Custom Dataset: composed of 3448 32x32 colour images organized in 4 classes: 3 non-overlapping classes and a mixed one, composed of pictures with objects of different categories present at the same time.

Network	Accuracy		Storage (Mb)			Time	
VGG-16	95.65%		56.14			22 min	
	Epoch 0	Epoch 10	Pre-M	AS/POD	PCE/FNN	Init	Train
AS+PCE (5)	29.03%	95.21%	2.12	3.12	0.02	2 min	10 min
AS+FNN (5)	94.63%	94.92%	2.12	3.12	0.0021	12.5 min	12 min
POD+FNN (5)	96.52%	96.66%	2.12	3.12	0.0021	28 sec	11.5 min
AS+PCE (6)	29.75%	95.79%	4.37	3.12	0.02	2.5 min	10 min
AS+FNN (6)	94.92%	95.36%	4.37	3.12	0.0021	12.5 min	12.5 min
POD+FNN (6)	96.23%	96.37%	4.37	3.12	0.0021	33 sec	13 min
AS+PCE (7)	28.59%	94.05%	6.62	0.78	0.02	1.5 min	11 min
AS+FNN (7)	94.34%	94.63%	6.62	0.78	0.0021	4.5 min	13 min
POD+FNN (7)	96.37%	96.52%	6.62	0.78	0.0021	33 sec	14 min

Future work and ideas

- Find a good criteria for determining the cut-off layer → Try to set up an iterative and adaptive procedure that converges to the optimal structure of the reduced network.
- Create a continuous version of the several filters in a CNN.
- Extend everything for the problem of object detection, thus to CNN with a more complex and deep architecture.
- General application of the described frameworks to other neural networks and layers, not only to convolutional layers..
- ...

Future Publications:

- L. Meneghetti, N. Demo, G. Rozza, ''**A reduced order model approach for Convolutional Neural Networks**'' (arXiv:2110.09163, soon submission to SIAM)
- Dedicated section inside Chapter 19 '' **A Deep Learning approach to improve ROM**'' of the **AROMA book**

References

- Chunfeng Cui, Kaiqi Zhang, Talgat Daulbaev, Julia Gusak, Ivan Oseledets, and Zheng Zhang. "**Active Subspace of Neural Networks: Structural Analysis and Universal Attacks**", (2020) SIAM Journal on Mathematics of Data Science (SIMODS)
- P. G. Constantine, E. Dow, and Q. Wang, '' **Active subspaces: Emerging ideas for dimension reduction in parameter studies**'', vol.2, SIAM, 2015
- Goodfellow, Y. Bengio, and A. Courville, ''**Deep Learning**'',MIT Press, 2016
- J. S. Hesthaven, G. Rozza, B. Stamm, et al., ''**Certified reduced basis methods for parametrized partial differential equations**'', vol. 590, Springer, 2016.
- Simonyan, Karen and Zisserman, Andrew, ''**Very deep convolutional networks for large-scale image recognition**'', In Y. Bengio and Y. LeCun, editors, 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.
- T. L. Fine, ''**Feedforward neural network methodology**'', Springer Science & Business Media, 2006.
- F. Romor, M. Tezzele, and G. Rozza, ''**ATHENA: Advanced Techniques for High dimensional parameter spaces to Enhance Numerical Analysis**'', Submitted, (2020)
- M. Tezzele, N. Demo, and G. Rozza, "**Shape optimization through proper orthogonal decomposition with interpolation and dynamic mode decomposition enhanced by active subspaces**", in *VIII International Conference on Computational Methods in Marine Engineering*, 2019.

THANK YOU FOR YOUR ATTENTION!

QUESTIONS?